Notice of Allowability	Application No.	Applicant(s)	
	09/538,132	LIAO, HENG	
	Examiner	Art Unit	
	Hussein A. El-chanti	2157	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *8/25/2006*.

2. ☒ The allowed claim(s) is/are *1-3,5,6,8-24 and 26-54*.

3. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☐ All b) ☐ Some* c) ☐ None of the:
      1. ☐ Certified copies of the priority documents have been received.
      2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
      3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.

5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
      1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
   (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☐ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. ☐ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date _____
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. ☐ Notice of Informal Patent Application
6. ☐ Interview Summary (PTO-413), Paper No./Mail Date _____ .
7. ☒ Examiner's Amendment/Comment
8. ☒ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____.

EXAMINER'S AMENDMENT

1. This action is responsive to amendment received on August 25, 2006

2. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Mr. George Yee on Sep. 21, 2006.

3. The application has been amended as follows:

1. (Currently amended): A method for classifying received network data in a network device comprising:

providing a language definition comprising regular expressions; [[and]]

parsing said regular expressions to identify grammatical structure among said regular expressions to produce a deterministic finite automaton (DFA); and

processing incoming network data with said language definition in accordance with a formal language processing technique including scanning said network data using lexical token scanning according to said language definition, said scanning including recognizing one or more lexical tokens contained in said network data using said DFA, said DFA including a representation of said lexical tokens,

said processing resulting in identification of protocol encapsulation formats in said network data

~~wherein said network data is treated as a stream of input bytes, said network data being organized into data packets, said scanning resulting in the identification of a data packet as belonging to one of a plurality of classes~~.

2. (Original): The method of claim 1 wherein said scanning includes identifying an arithmetic operation and performing said arithmetic operation.

3. (Currently amended): The method of claim 1 wherein said network data is treated as a stream of input bytes and~~wherein~~ said scanning includes identifying a skip operation and in response thereto skipping over one or more subsequent input bytes.

4. (Canceled)~~: The method of claim 1 wherein said lexical scanning includes providing a set of regular expressions, each regular expression having an associated class identifier.~~

5. (Currently amended): The method of claim 1 further ~~including providing a deterministic finite automaton (DFA) comprising plural states,~~ wherein said DFA comprises plural states and said network data is organized into data packets, said step of scanning including recognizing data packets using said DFA including transitioning from one state to another.

6. (Original): The method of claim 5 wherein said data packets are variable length data packets.

7. (Canceled): ~~The method of claim 5 wherein said DFA is defined by a set of regular expressions.~~

8. (Currently amended): The method of claim [[7]]6 further including generating a grammar tree data structure representative of said regular expression, producing a non-deterministic finite automaton (NFA) from said grammar tree data structure, and converting said NFA to produce said DFA.

9. (Original): The method of claim 5 wherein some of said states include one or more associated computer instructions and wherein said computer instructions are executed in connection with transitioning to a state.

10. (Original): The method of claim 9 wherein some of said states further include a skip instruction.

11. (Currently amended): In a network data switching device, a method for classifying data packets comprising steps of:

providing a language definition in the form of one or more regular expressions, each having an associated class identifier indicative of a protocol encapsulation format;

parsing said regular expressions to identify grammatical structure among said regular expressions to produce a deterministic finite automaton (DFA), said DFA including a representation of a plurality of lexical tokens;

receiving plural data packets, each having a length not necessarily equal to one another; and

for each data packet, processing said data packet in accordance with a formal language processing technique including determining a matching regular expression from among said regular expressions that matches said data packet, wherein said each data packet is classified according to the class identifier associated with said matching regular expression,

said determining comprising lexical token scanning of said data packet to identify said lexical tokens using said DFA.

12. (Original): The method of claim 11 wherein said data packets comprise a data stream and said determining includes lexically scanning said data stream.

13. (Canceled): ~~The method of claim 11 wherein said regular expressions are represented by a deterministic finite automaton (DFA).~~

14. (Currently amended): The method of claim [[13]]11 wherein said DFA is in compressed form.

15. (Original): The method of claim 11 further including compiling said regular expressions to produce said DFA.

16. (Original): The method of claim 15 wherein said compiling produces a non-deterministic finite automaton (NFA) as intermediate data structure, said compiling further includes converting said NFA to produce said DFA.

17. (Original): The method of claim 16 further including reducing said DFA to a compressed form.

18. (Original): The method of claim 11 wherein said data packet comprises plural bytes, and said determining includes detecting an operator indicating a number of bytes to be skipped.

19. (Original): The method of claim 18 wherein said number is specified by the value of a current input byte.

20. (Original): The method of claim 18 wherein said number is specified in a register.

21. (Original): The method of claim 18 wherein said determining further includes detecting an operator indicating a value to be saved in a register.

22. (Original): The method of claim 21 wherein said determining further includes detecting an operator indicating a logical or mathematical operation to be performed on the contents of said register.

23. (Currently amended): In a data packet receiving and forwarding device, a method for classifying received data packets constituting a data stream ~~comprising a stream of data~~, said method comprising steps of:

receiving a description of classification rules in the form of a classification language definition, said classification language definition including regular expressions;

compiling said classification language definition to produce a deterministic finite automaton (DFA) comprising plural states, including parsing said regular expressions to identify grammatical structure among said regular expressions, said DFA including a representation of a plurality of lexical tokens;

configuring a programmable hardware packet classifier with said DFA; and

receiving said data stream and processing said data stream in accordance with a formal language processing technique including scanning said data stream with said hardware packet classifier to classify said received data packets into one of a plurality of protocol encapsulation formats, said scanning comprising using said DFA to identify said lexical tokens in said data stream.

24. (Original): The method of claim 23 wherein said compiling includes associating arithmetic and logic instructions with some of said states.

25. (Canceled)

26. (Currently amended): The method of claim [[25]]23 wherein said regular expressions include arithmetic and logic operations.

27. (Original): The method of claim 26 wherein said regular expressions further include skip operations.

28. (Original): The method of claim 27 wherein said regular expressions further include data storage operations.

29. (Original): The method of claim 23 wherein said DFA is in compressed format.

30. (Previously presented): The method of claim 23 further including:

receiving a second description of classification rules in the form of a second classification language definition;

compiling said second classification language definition to produce a second DFA;

configuring a programmable hardware packet classifier with said second DFA; and

applying said data stream to said hardware packet classifier to classify said received data packets,

wherein said data packets are classified according to said second classification rules, thereby facilitating changing packetizing policies in said data packet routing device.

31. (Currently amended): A network data packet classifier comprising:

an input port for receiving network data packets comprising a stream of data;

a memory assemblage configured with data representing a deterministic finite automaton (DFA), said DFA produced from a language definition comprising plural regular expressions by parsing said regular expressions to identify grammatical

structure among said regular expressions ~~representing plural regular expressions according to a language definition~~; and

decompression logic operatively coupled to said memory assemblage and configured to process said stream of data using said language definition in accordance with a formal language processing technique including scanning said stream of data with said DFA to find a matching one of said regular expressions,

said regular expressions having corresponding class identifiers, each class identifier corresponding to a protocol encapsulation format,

said scanning comprising using said DFA to identify lexical tokens in said stream of data, said DFA including a representation of said lexical tokens,

wherein each of said network data packets is associated with the class identifier of said regular expression that matches said each network data packet.

32. (Original): The classifier of claim 31 wherein some of said regular expressions include arithmetic instructions and logic instructions, said memory assemblage further configured to contain said instructions, the classifier further including an arithmetic logic unit operatively coupled to said decompression logic and configured to execute said instructions.

33. (Original): The classifier of claim 32 further including at least one register operatively coupled to said arithmetic logic unit, said arithmetic logic unit further configured to store data into said register in response to a save instruction.

34. (Original): The classifier of claim 32 further including skip logic operatively coupled to said logic component and configured to skip over an amount of data in response a skip instruction.

35. (Original): The classifier of claim 31 wherein said network data packets can vary from one packet to another.

36. (Original): The classifier of claim 31 wherein said DFA is in compressed form.

37. (Original): The classifier of claim 36 wherein said DFA comprises plural non-default states and plural default states, and said memory assemblage comprises a base memory, a next-state memory, and a default-state memory; said base memory configured to contain address locations of said next-state memory, said next-state memory representing all of said non-default states, said default-state memory representing all of said default states.

38. (Original): The classifier of claim 37 wherein said memories are random access memories.

39. (Original): The classifier of claim 37 wherein said memories are read-only memories.

40. (Currently amended): A network data packet classifier comprising:

an input configured to provide a data packet comprising a stream of data;

a first system of memory configured with data representing a deterministic finite automaton (DFA), said DFA defined in accordance with a language definition comprising a set of regular expressions and produced by parsing said set of regular expressions to identify grammatical structure among said regular expressions, said DFA comprising plural states including an initial state and plural terminating states;

a system of logic circuits operatively coupled to said first system of memory and to said input, and configured to process said ~~data~~ stream of data using said language definition in accordance with a formal language processing technique including a step to lexically scan said data stream with said DFA to produce a reached terminating state which corresponds to recognition of a lexical token; and

a second system of memory configured with data representing a class index corresponding to each of said terminating states and configured to output a class index in response to the production of said reached terminating state, said class index corresponding to a protocol encapsulation format.

41. (Original): The classifier of claim 40 further including a third system of memory configured to contain current state information for plural input channels, said system of logic circuits operatively coupled to said third system of memory to initialize said DFA in accordance with current state information corresponding to the input channel associated with said data packet.

42. (Original): The classifier of claim 40 wherein some of said states have one or more associated instructions, the classifier further including an arithmetic

logic unit operatively coupled to said system of logic circuits and configured to execute said instructions.

43. (Original): The classifier of claim 42 further including at least one register operatively coupled to said arithmetic logic unit, said arithmetic logic unit further configured to store data into said register in response to a save instruction.

44. (Original): The classifier of claim 42 further including skip logic operatively coupled to said logic component and configured to skip over an amount of data in response a skip instruction.

45. (Original): The classifier of claim 40 wherein said stream of data is a stream of bytes.

46. (Original): The classifier of claim 40 wherein said data packets vary from one packet to another.

47. (Currently amended): A network packet classifier comprising:

means for receiving an incoming network packet;

means for processing said network packet in ~~accordance~~ using a language definition in accordance with a formal language processing technique including means for classifying said network packet by matching the pattern of constituent data of said network packet against plural regular expressions, each regular expression having a corresponding class identifier; and

means for outputting a class identifier of the regular expression which matches said network packet, said class identifier corresponding to a protocol encapsulation format,

said means for classifying including means for lexical token scanning to recognize one or more lexical tokens in said network packet using a deterministic finite automaton (DFA), said DFA including a representation of said one or more lexical tokens, said DFA produced by parsing said regular expressions to identify grammatical structure among said regular expressions.

48. (Currently amended): The classifier of claim 47 wherein said means for classifying includes a memory component configured with data to represent said DFA ~~a deterministic finite automaton (DFA)~~.

49. (Original): The classifier of claim 48 wherein said means for outputting includes a second memory component configured with said class identifiers.

50. (Original): The classifier of claim 47 wherein said regular expressions include arithmetic specifiers and said means for classifying includes an arithmetic logic unit configured to perform operations in accordance with said arithmetic specifiers.

51. (Currently amended): A network packet classifier comprising:

a dual-ported memory component;

first classification logic operatively coupled to a first port of said dual-ported memory component and having a first input for receiving a data stream; and

second classification logic operatively coupled to a second port of said dual-ported memory component and having a second input for receiving a data stream,

said memory component configured to contain a deterministic finite automaton (DFA) representative of a language definition and comprising plural states,

said DFA representing plural regular expressions of said language definition for matching data packets and produced by parsing said regular expressions to identify grammatical structure among said regular expressions,

said first and second classification logic each configured to process an associated data stream using said language definition according to a formal language processing technique including a step to scan said associated data stream using said DFA to identify data packets contained therein using said DFA to perform lexical token scanning of said data packets to produce plural lexical tokens, said DFA including a representation of said lexical tokens ~~and~~ to classify identified data packets into one of a plurality of protocol encapsulation formats.

52. (Original): The classifier of claim 51 wherein said data packets are characterized in being variable in length.

53. (Original): The classifier of claim 51 wherein said regular expressions include arithmetic and logic operators.

54. (Original): The classifier of claim 51 wherein said regular expressions include a skip operator.

Reasons for Allowance

4. Claims 1-3, 5-6, 8-24, 26-54 are allowed.

5. The following is an examiner's statement of reasons for allowance:

The prior art of record does not teach neither singly or in combination the claimed limitation "parsing said regular expressions to identify grammatical structure among said regular expressions to produce a deterministic finite automaton (DFA); and processing incoming network data with said language definition in accordance with a formal language processing technique including scanning said network data using lexical token scanning according to said language definition, said scanning including recognizing one or more lexical tokens contained in said network data using said DFA, said DFA including a representation of said lexical tokens, said processing resulting in identification of protocol encapsulation formats in said network data" as in claims 1-3, 5-6, 8-24, 26-54.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Hussein A. El-chanti whose telephone number is (571)272-3999. The examiner can normally be reached on Mon-Fri 8:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ario Etienne can be reached on (571)272-4001. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Hussein El-chanti

Sep. 27, 2006